

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : **06-020026**  
 (43)Date of publication of application : **28.01.1994**

~~(51)Int.Cl.:~~

**G06F 15/62**  
**G06F 15/62**  
**B41B 23/00**  
**B41B 27/00**  
**B41J 2/485**  
**G06F 15/20**

**(21)Application number : 04-174410**  
**(22)Date of filing : 01.07.1992**

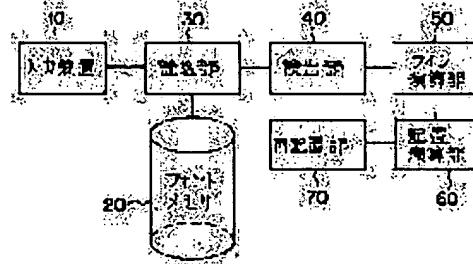
**(71)Applicant : FUJI XEROX CO LTD**  
**(72)Inventor : MURAMATSU MASAHIKO**

## **(54) AUTOMATIC CHARACTER PATTERN CORRECTING DEVICE**

### **(57)Abstract:**

**PURPOSE:** To provide an automatic character pattern correcting device which can correct the positions of character patterns of the alphanumeric, HIRAGANA (cursive form of Japanese syllabary) characters, etc., to each optimum position.

**CONSTITUTION:** A font memory 20 stores the metrics information to show the positional relation between the alphanumeric character pattern corresponding to a character code and the circumscribed square of the alphanumeric characters corresponding to the original point of the alphanumeric character pattern. A reading part 30 reads the alphanumeric character pattern corresponding to the character code given from an input device 10 and the metrics information on the alphanumeric character pattern out of the memory 20 and transfers them to a detecting means (part) 40. The means 40 detects the positions of the alphanumeric characters based on the alphanumeric character pattern and its metrics information given from the part 30. The arithmetic means (a line arithmetic part 50 and an arrangement arithmetic part 60) calculate the positions of the alphanumeric characters to be rearranged based on the metrics information. A rearrangement means (part) 70 rearranges the alphanumeric characters based on the detecting result of the means 40 and the arithmetic results of the arithmetic means.



### **LEGAL STATUS**

[Date of request for examination] **28.06.1999**  
 [Date of sending the examiner's decision of rejection] **11.12.2001**

[Kind of final disposal of application other than the examiner's decision of rejection or application]

[converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of  
rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-20026

(43)公開日 平成6年(1994)1月28日

(51)Int.Cl.<sup>5</sup>

G 0 6 F 15/62

識別記号 庁内整理番号

3 2 5 D 8125-5L

F I

技術表示箇所

4 1 0 Z 9287-5L

B 4 1 B 23/00

27/00

8804-2C

B 4 1 J 3/12

B

審査請求 未請求 請求項の数2(全16頁) 最終頁に続く

(21)出願番号

特願平4-174410

(22)出願日

平成4年(1992)7月1日

(71)出願人 000005496

富士ゼロックス株式会社

東京都港区赤坂三丁目3番5号

(72)発明者 村松 正彦

神奈川県川崎市高津区坂戸3丁目2番1号

K S P R & D ビジネスパークビル

富士ゼロックス株式会社内

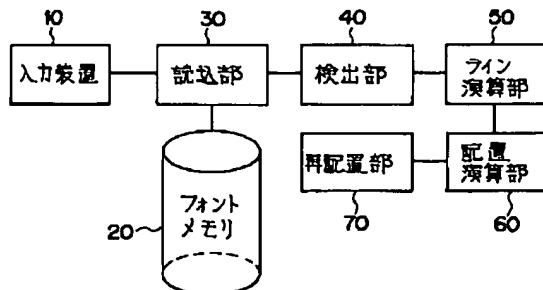
(74)代理人 弁理士 木村 高久

(54)【発明の名称】 文字パターン自動修正装置

(57)【要約】

【目的】英数文字、ひら仮名などの文字パターンの位置を最適な位置に修正することのできる文字パターン自動修正装置を提供する。

【構成】フォントメモリ20には、文字コードに対応する英数文字パターン及び該英数文字パターンの原点に対する当該英数文字の外接四角形の位置関係を示すメトリクス情報が格納されている。読み込部30は、入力装置10からの文字コードに対応する英数文字パターン及びそのメトリクス情報をフォントメモリ20から読み出して、検出手段(検出部40)に渡す。検出手段は、読み込部30から渡された英数文字パターン及びそのメトリクス情報から、当該英数文字の位置を検出する。演算手段(ライン演算部50及び配置演算部60)は、上記メトリクス情報から再配置すべき英数文字の位置を演算する。再配置手段(再配置部70)は、上記検出手段による検出結果及び上記演算手段による演算結果に基づいて、当該英数文字を再配置する。



## 【特許請求の範囲】

【請求項1】英数文字が表現された英数文字パターンの原点に対する当該英数文字の位置を修正する文字パターン自動修正装置であって、

前記英数文字パターンと、該英数文字パターンの原点に対する当該英数文字の外接四角形の位置関係を示すメトリクス情報とから、当該英数文字の位置を検出する検出手段と、

前記メトリクス情報から再配置すべき前記英数文字の位置を演算する演算手段と、

前記検出手段による検出結果及び前記演算手段による演算結果に基づいて、前記英数文字を再配置する再配置手段とを具えたことを特徴とする文字パターン自動修正装置。

【請求項2】文字が表現された文字パターンの原点に対する当該文字の位置を修正する文字パターン自動修正装置であって、

標準文字が表現された標準文字パターンと、該標準文字パターンの原点に対する当該標準文字の外接四角形の位置関係を示すメトリクス情報を記憶する記憶手段と、

前記文字パターン及び該文字パターンの原点に対する当該文字の外接四角形の位置関係を示すメトリクス情報と、前記記憶手段の記憶内容に基づいて、当該文字の位置及びその文字に対応する標準文字の位置を検出する検出手段と、

前記検出手段による検出結果に基づいて前記文字の移動量を演算する演算手段と、

前記演算手段による演算結果に基づいて前記文字の再配置を行う再配置手段とを具えたことを特徴とする文字パターン自動修正装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】この発明は、文字の位置を自動修正する文字パターン自動修正装置に関する。

## 【0002】

【従来の技術】従来においては、ユーザ側で新規に文字パターンを作成しシステムに登録することにより、その文字を使用することが可能になっている。その文字パターンは、ユーザによってベースライン等の位置や文字並びも考慮されて作成されるが、人為的なミスも多い。また文字並びを全く考慮せずに文字パターンが作成されている場合もある。

【0003】このような問題を解決すべく、文字パターンを検査するものとしては、特開昭60-29783号公報、特開平3-176764号公報に開示されたものが知られている。

【0004】特開昭60-29783号公報に開示されたものは、作成文字パターンと標準文字パターンとを同一画面上に並べて表示するか、或いは同一紙面上に並べて印刷して、両文字パターンを目視で検査するようにし

たものである。

【0005】また特開平3-176764号公報に開示されたものは、アルファベットの大文字を構成するドットパターンの最下部のライン位置を検出することにより文字画像からベースラインの位置を検出し、この検出した位置と基準値とを比較することにより、アルファベットの大文字についてのベースラインが正しいか否かを判定するようにしたものである。

## 【0006】

【発明が解決しようとする課題】しかしながら、上記特開昭60-29783号公報に開示されたものでは、デザイン的な字体検査のみに限定しているため、ベースラインやCap height、X height等の位置検査には向きであった。

【0007】また、上記特開平3-176764号公報に開示されたものでは、オーバハング(overhang)と呼ばれる視覚補正のためのデザイン的な手法による調整量を考慮していないので、正確な文字パターンを得ることができない。またアルファベットの大文字についての検査は可能であるが、アルファベットの小文字、数字、ひら仮名、カタカナなどの文字についてはチェックすることができない。すなわち、アルファベットの大文字のドットパターンのチェックに限定しているため、JISのような7000文字近くもある文字パターンの位置をチェックすることは不可能である。

【0008】この発明は、英数文字、ひら仮名などの文字パターンの位置を最適な位置に修正することのできる文字パターン自動修正装置を提供することを目的とする。

## 【0009】

【課題を解決するための手段】上記目的を達成するため、第1の発明は、英数文字が表現された英数文字パターンの原点に対する当該英数文字の位置を修正する文字パターン自動修正装置であって、前記英数文字パターンと、該英数文字パターンの原点に対する当該英数文字の外接四角形の位置関係を示すメトリクス情報とから、当該英数文字の位置を検出する検出手段と、前記メトリクス情報から再配置すべき前記英数文字の位置を演算する演算手段と、前記検出手段による検出結果及び前記演算手段による演算結果に基づいて、前記英数文字を再配置する再配置手段とを具えている。

【0010】また第2の発明は、文字が表現された文字パターンの原点に対する当該文字の位置を修正する文字パターン自動修正装置であって、標準文字が表現された標準文字パターンと、該標準文字パターンの原点に対する当該標準文字の外接四角形の位置関係を示すメトリクス情報を記憶する記憶手段と、前記文字パターン及び該文字パターンの原点に対する当該文字の外接四角形の位置関係を示すメトリクス情報と、前記記憶手段の記憶内容に基づいて、当該文字の位置及びその文字に対応す

る標準文字の位置を検出する検出手段と、前記検出手段による検出結果に基づいて前記文字の移動量を演算する演算手段と、前記演算手段による演算結果に基づいて前記文字の再配置を行う再配置手段とを具えている。

【0011】

【作用】第1の発明の文字パターン自動修正装置では、検出手段によって、英数文字パターン及び該英数文字パターンの原点に対する当該英数文字の外接四角形の位置関係を示すメトリクス情報から当該英数文字の位置が検出され、また演算手段によって、前記メトリクス情報から再配置すべき英数文字の位置が演算され、更に再配置手段によって、前記検出手段による検出結果及び前記演算手段による演算結果に基づいて、前記英数文字が再配置される。従って、アルファベットの大文字及び小文字や数字の英数文字パターンの位置を、最適な位置に自動修正することができる。

【0012】また、第2の発明の文字パターン自動修正装置では、記憶手段には、標準文字が表現された標準文字パターン及び該標準文字パターンの原点に対する当該標準文字の外接四角形の位置関係を示すメトリクス情報が記憶されており、検出手段によって、前記文字パターン及び該文字パターンの原点に対する当該文字の外接四角形の位置関係を示すメトリクス情報と、前記記憶手段の記憶内容とに基づいて、当該文字の位置及びその文字に対応する標準文字の位置が検出され、また演算手段によって、前記検出手段による検出結果に基づいて前記文字の移動量が演算され、更に再配置手段によって、前記演算手段による演算結果に基づいて前記文字が再配置される。従って、アルファベットの大文字及び小文字や数字、ひら仮名文字などのJISの文字コードに対応する文字パターンの位置を、最適な位置に自動修正することができる。

【0013】

【実施例】以下、本発明の第1の実施例及び第2の実施例について添付図面を参照して説明する。

【0014】最初に第1の実施例について、図1乃至図9を参照して説明する。

【0015】図1は本発明に係る文字パターン自動修正装置の一実施例を機能ブロック図で示したものである。

【0016】同図において、文字パターン自動修正装置は、文字フォントに対応する文字コードを入力する入力装置10と、英数文字パターンを格納しているフォントメモリ20と、指定された文字コードに対応する文字パターンをフォントメモリ20から読み込む読込部30と、各文字の位置を検出する検出部40と、英数文字の並び線（ライン）を求めるライン演算部50と、検出された位置と移動すべき位置とを求める誤差（移動

$$\text{キャップ・オーバハング} = \text{ラインLN1の値} - \text{ラインLN2の値} \dots (1)$$

$$\text{ミーン・オーバハング} = \text{LN3の値} - \text{LN4の値} \dots (2)$$

$$\text{ベース・オーバハング} = \text{LN5の値} - \text{LN6の値} \dots (3)$$

量）を演算する配置演算部60と、配置演算部60により求められた移動量に基づいて文字を再配置する再配置部70とを有して構成されている。

【0017】ここで、フォントメモリ20には、ASC II（アスキ）の文字コードと、その文字コードに対応する英数文字パターン（アウトライン又はピットマップ）及びメトリクス情報からなるパターン情報とが格納されている。

【0018】このパターン情報について、図2に示す“7”という1つのアウトラインを例にとって説明する。

【0019】メトリクス情報は、“7”というアウトラインを表現するための各直線の交点である6つの制御点Cと、図2中点線で囲まれたバウンディングボックス（文字パターンに接する最大矩形枠）情報と、文字の送り幅であるエスケイプメントWの情報とを有して構成されている。バウンディングボックス情報は、原点（0,0）からのオフセット値であるアセンダーA（正の値）、ディセンダーD（正の値）、ライトベアリングR（正の値）及びレフトベアリングL（負の値）の4つの情報から構成されている。

【0020】また英数文字には原則的な並び線（ライン）が幾つかあって、各文字はそれらの線に沿って並んでいるのが普通である。それらの各ラインについて図3を参照して説明する。

【0021】この実施例では、8つのラインLN1、LN2、LN3、LN4、LN5、LN6、LN7、LN8が設定されている。

【0022】LN1はアセンダーライン（ascender line）を示し、LN2はキャップライン（cap line）を示し、LN4はミーンライン（mean line）を示し、LN5はベースライン（base line）を示し、LN7はディセンダーライン（descender line）を示している。ここでアセンダーラインLN1とはアルファベットの小文字の最も高いラインのことであり、ディセンダーラインLN7とはアルファベットの小文字の最も低いラインのことである。

【0023】ラインLN3、LN6、LN8はオーバハング（overhang）を求めるために必要なラインである。ここでオーバハングとは視覚補正のためのデザイン的な手法による調整のことである。このオーバハングには、キャップ・オーバハング（cap overhang）、ミーン・オーバハング（mean overhang）、ベース・オーバハング（base overhang）及びディセンダー・オーバハング（descender overhang）の4種類があり、これらは次のようにして求めることができる。

【0024】

## ディセンダー・オーバハンギング=LN7の値-LN8の値

… (4)

これらのオーバハンギングはライン演算部50によって演算される。

【0025】この実施例においては、ベース・ラインLN5の値はY座標値=0としているので、アセンダー・ラインLN1、キャップ・ラインLN2、ラインLN3、ミーン・ラインLN4、及びベース・ラインLN5の各値は正の値となり、これに対し、ラインLN6、ディセンダー・ラインLN7及びラインLN8の各値は負の値となる。

【0026】なおこの実施例では、上述した検出手段は検出部40を有して構成されており、上述した演算手段はライン演算部50及び配置演算部60を有して構成されており、上述した再配置手段は再配置部70を有して構成されている。

【0027】次に具体例を挙げて文字パターンの位置修正を説明する。

【0028】入力装置10から、文字“A”、“n”、“1”、“O”の各文字にそれぞれ対応する文字コードが順次入力されると、読出部20は、フォントメモリ20からフォントファイルを読み込み、そのファイル内から、それらの文字コードに対応する文字パターンを読み込む。この結果、図4に示すような英数文字パターンの並びが得られることとなる。しかし、これらの英数文字パターンは、図4に示すように最適な並びでないということが分かる。なおこの実施例では、前提条件として英数文字パターンのデザインは正しいものとする。

【0029】そこで、それらの文字パターンの位置を検査し、最適な位置に修正する必要があるので、次に、本発明に係る文字パターンの位置の検査及び修正について説明する。

【0030】ここでは、最初の文字“A”を例にとって説明することにし、その文字の位置を検査し修正する処理過程を図5を用いて説明する。

【0031】読込部30でフォントファイルを読み込み、文字コードによって文字を判別する。この場合、ASCIIコードになっているため大文字の“A”はOx41である。すなわち、読込部30は、入力装置10からの文字コードOx41を受け取ると、大文字の“A”であると判定する。

【0032】ここで、大文字の“A”的パターンは図5(a)に示すような内容であり、そのバウンディングボックスは、エスケイプメントWの中心より左側(紙面に向かって左側)に存在し、かつバウンディングボックスの底辺がディセンダーラインに接した位置に存在しているものとする。

【0033】検出部40では、大文字“A”的バウンディングボックス情報をから、その重心位置を検出する。ここでは、バウンディングボックスの対角線の交点を重心位置とする。具体的には、バウンディングボックスは、

10

20

30

40

50

ディセンダー・ラインにおけるライトペアリングの値及びレフトペアリングの値でそれぞれ示される点、及びアセンダー・ラインにおけるライトペアリングの値及びレフトペアリングの値でそれぞれ示される点を頂点とする矩形領域で表現することができる。従って、これらの値に基づいて対角線の交点を求めることができる。

【0034】ライン演算部50では、図3に示した様な各ラインを求める(この演算の詳細については後述する)。

【0035】配置演算部60では、重心位置を示す座標値(X0、Y0)がエスケイプメントWの中心に存在するか否かを調べる。中心に存在しなければ、X方向の移動量を計算する。この場合、移動量XM=W-X0となる。

【0036】この様にX方向の移動量を演算したならば、配置演算部60では、次にY方向の移動量を計算する。“A”的場合には、図3に示したベースラインLN5がバウンディングボックスの底辺と接した状態が適切な位置であるので、この文字に対するディセンダーの値は「0」である。しかし、この例では、バウンディングボックスの底辺がディセンダーラインに接しているので、この場合のY方向の移動量はディセンダーDの値である。

【0037】再配置部70では、X方向にXM、Y方向にdだけ移動する。これにより、バウンディングボックスは、図5(b)に示すようにX方向にXMだけ移動され、さらに、図5(b)に示すようにY方向にディセンダーDの値だけ移動される。なお移動量に基づくX方向、Y方向への移動の際はその順序は問わない。

【0038】上述したように大文字“A”的ようにバウンディングボックスの底辺がベースラインに接する他の文字については同様の処理で文字の位置を修正することができる。

【0039】このような文字以外の文字についても、X方向の移動量については上記同様の方法で求めることができ、一方Y方向の移動量については、バウンディングボックスの底辺がラインLN5、LN6、LN7、LN8のいずれかと接するのかが分かれば求めることができる。

【0040】一般的に、アルファベット{A、B、D、E、F、H、I、K、L、M、N、P、R、T、V、W、X、Y、Z}及び数字{1、2、4、7}は共に、バウンディングボックスの底辺が図3に示したベースラインLN5と接する。

【0041】またアルファベット{C、G、O、Q、S、U}及び数字{0、3、6、8、9}は共に、バウンディングボックスの底辺が図3に示したラインLN6と接する。

【0042】これに対し、アルファベットの小文字は、

高さとラインの並びによってショートレター、アセンダーレター、ディセンダーレター、ロングレターの4種類に分類することができるので、その種類に応じた配置となる。

【0043】ショートレターのうち、{a, c, e, o, s, u}は共にパウンディングボックスの底辺が図3に示したラインLN6と接し、また{m, n, r, v, w, x, z}は共にパウンディングボックスの底辺が図3に示したベースラインLN5と接する。

【0044】アセンダーレターのうち、{b, d, t}は共にパウンディングボックスの底辺が図3に示したラインLN6と接し、また{h, i, k, l}は共にパウンディングボックスの底辺が図3に示したベースラインLN5と接する。

【0045】ディセンダーレターのうち、{p, q}は共にパウンディングボックスの底辺が図3に示したディセンダーラインLN7と接し、また{g, y}は共にパウンディングボックスの底辺が図3に示したラインLN8と接する。

【0046】ロングレター{j}はパウンディングボックスの底辺が図3に示したラインLN8と接する。

【0047】この実施例においては、アルファベットの小文字は上述した規則に従うものとする。

【0048】次にライン演算部50によるベースライン以外の各ラインの求め方について、図6及び図7を参照して説明する。

【0049】ここで前提条件として、入力するフォントには文字“H”、“O”、“x”、“p”が必ずあるものとする。なお、入力する文字はこれらの文字に限定されるものではなく、文字“H”と同様にパウンディングボックスの底辺がベースラインLN5と接する文字、文字“O”と同様にパウンディングボックスの底辺がラインLN6と接する文字、ショートレタ“x”と同様にパウンディングボックスの底辺がベースラインLN5と接する文字、ディセンダーレター“p”と同様にパウンディングボックスの底辺がディセンダーラインLN7と接する文字にそれぞれ対応する文字を入力するようにしても良い。

\*

$$\text{overhang} = (\text{"O"} \text{ の } \text{BBox} \text{ の 高さ} - \text{"H"} \text{ の } \text{BBox} \text{ の 高さ}) / 2$$

… (5)

この式(5)を演算することにより求められたオーバハングの値がキャップ・オーバハング及びベース・オーバハングとなる。

【0058】上記ステップ605において大文字の“O”的文字コードでない場合は、小文字の“x”的文字コードか否かを判断する(ステップ609)。

【0059】小文字の“x”的文字コードである場合は、小文字の“x”的メトリクス情報を読み込んで(ステップ610)、上記同様にして、その文字のパウンディングボックスBBoxの高さを求める(ステップ6150)

\* 【0050】なお図6及び図7において、“BBox”はパウンディングボックスを示し、“overhang”はオーバハングを示し、“cap overhang”はキャップ・オーバハングを示し、“mean overhang”はミーン・オーバハングを示し、“base overhang”はベース・オーバハングを示し、“descender overhang”はディセンダー・オーバハングを示している。

【0051】最初に、overhang (cap overhang=mean overhang =base overhang =descender overhangと仮定する)を求める。この演算処理について、図6に示すフローチャートを参照して説明する。

【0052】ライン演算部50は、文字コードを読み込んで(ステップ601)、大文字の“H”的文字コードか否かを判断する(ステップ602)。大文字の“H”的文字コードである場合は、大文字の“H”的メトリクス情報を読み込んで(ステップ603)、その文字のパウンディングボックスBBoxの高さを求める(ステップ604)。

【0053】そのパウンディングボックスBBoxの高さはアセンダーの値とディセンダーの値とを加算することにより得られる。このアセンダーの値及びディセンダーの値は、今読み込んだ大文字の“H”的メトリクス情報に含まれているので、そのメトリクス情報から容易に知ることができる。

【0054】また上記ステップ602において大文字の“H”的文字コードでない場合は、次に大文字の“O”的文字コードか否かを判断する(ステップ605)。

【0055】大文字の“O”的文字コードである場合は、大文字の“O”的メトリクス情報を読み込んで(ステップ606)、その文字のパウンディングボックスBBoxの高さを求める(ステップ607)。この場合のパウンディングボックスBBoxの高さも上記同様にして求めることができる。

【0056】このようにして大文字の“H”及び“O”的パウンディングボックスBBoxの高さを求めたならば、ライン演算部50は、次式を演算することによりオーバハングを求める(ステップ608)。

【0057】

1).

【0060】一方、ステップ609において小文字の“x”的文字コードでない場合は、小文字の“p”的文字コードか否かを判断する(ステップ612)。

【0061】小文字の“p”的文字コードの場合は、小文字の“p”的メトリクス情報を読み込み(ステップ613)、その後、ステップ611に進む。

【0062】なおステップ612において小文字の“p”的文字コードでない場合はステップ601に戻りこのステップ以降を実行する。

【0063】次にライン演算部50による上記各ラインの決定処理について、図7に示すフローチャートを参照して説明する。

【0064】最初に、ライン演算部50は、`cap overhang=mean overhang=base overhang=descender overhang`と定義し（ステップ701）、この定義内容と、予め設定されたベース・ラインの値に基づいて、ラインLN1、LN2、LN3、LN4、LN6、LN7、LN8を求める。具体的には次の様にして各ラインを求めるようにしている。

【0065】最初に、ベース・ラインLN5の値からベース・オーバハングの値を減算することにより、ラインLN6を求めることができる（ラインLN6=ベース・ラインLN5の値-ベース・オーバハングの値）（ステップ702）。このラインの値は負の値となる。

【0066】次にベース・ラインLN5の値と“H”のBBoxの高さとを加算することにより、キャップ・ラインLN2を求めることができる（キャップ・ラインLN2=ベース・ラインLN5の値+“H”のBBoxの高さ）（ステップ703）。このラインの値は正の値となる。

【0067】こうして得られたキャップ・ラインLN2の値とキャップ・オーバハングの値とを加算することにより、アセンダー・ラインLN1を求めることができる（アセンダー・ラインLN1=キャップ・ラインLN2の値+キャップ・オーバハングの値）（ステップ704）。このラインの値は正の値となる。

【0068】ベース・ラインLN5の値と小文字“x”的BBoxの高さと加算することにより、ミーン・ラインLN4を求めることができる（ミーン・ラインLN4=ベース・ラインLN5の値+“x”的BBoxの高さ）（ステップ705）。このラインの値は正の値となる。

【0069】こうして求められたミーン・ラインLN4の値とミーン・オーバハングの値とを加算することにより、ラインLN3を求めることができる（ラインLN3=ミーン・ラインLN4の値+ミーン・オーバハングの値）（ステップ706）。このラインの値は正の値となる。

【0070】また、こうして求められたLN3の値から、小文字“p”的BBoxの高さを減算することにより、ディセンダー・ラインLN7を求めることができる（ディセンダー・ラインLN7=ラインLN3の値-“p”的BBoxの高さ）（ステップ707）。このラインの値は負の値となる。

【0071】さらに、こうして求められたディセンダー・ラインLN7の値から、ディセンダー・オーバハングの値を減算することにより、ラインLN8を求めることができる（ラインLN8=ディセンダー・ラインLN7の値-ディセンダー・オーバハングの値）（ステップ7

08）。このラインの値は負の値となる。

【0072】以上の処理でライン演算部50により各ラインが決定されたので、それらの各ラインの値に基づいて、各文字のY方向の移動量を演算することができる。

【0073】次に、配置演算部60による各文字のY方向の移動量の演算処理について、図8に示すフローチャートを参照して説明する。

【0074】配置演算部60は、文字コードを読み込み（ステップ801）、次に示す文字群中のいずれかの文字に対応する文字コードであるか否かを判断する（ステップ802）。

【0075】すなわち、アルファベットの大文字{A、B、D、E、F、H、I、K、L、M、N、P、R、T、V、W、X、Y、Z}、数字{1、2、4、7}、アルファベットの小文字{m、n、r、v、w、x、z、h、i、k、l}のいずれかの文字の文字コードであるか否かを判断する。

【0076】ここで、いずれかの文字コードである場合は、その文字のパウンディングボックスBBoxの底辺がベース・ラインLN5に接するように、Y方向への移動量を求める（ステップ803）。

【0077】ステップ803を終了した場合、ステップ802において該当する文字の文字コードでない場合は、次に示す文字群中のいずれかの文字に対応する文字コードであるか否かを判断する（ステップ804）。

【0078】すなわち、アルファベットの大文字{C、G、O、Q、S、U}、数字{0、3、6、8、9}及びアルファベットの小文字{a、c、e、o、s、u、b、d、t}のいずれかの文字の文字コードであるか否かを判断する。

【0079】ここで、いずれかの文字コードである場合は、その文字のパウンディングボックスBBoxの底辺がラインLN6に接するように、Y方向への移動量を求める（ステップ805）。

【0080】ステップ805を終了した場合、ステップ804において該当する文字の文字コードでない場合は、次に示す文字群中のいずれかの文字に対応する文字コードであるか否かを判断する（ステップ806）。

【0081】すなわち、アルファベットの小文字{p、q}のいずれかの文字の文字コードであるか否かを判断する。

【0082】ここで、いずれかの文字コードである場合は、その文字のパウンディングボックスBBoxの底辺がディセンダー・ラインLN7に接するように、Y方向への移動量を求める（ステップ807）。

【0083】ステップ807を終了した場合、ステップ806において該当する文字の文字コードでない場合は、次に示す文字群中のいずれかの文字に対応する文字コードであるか否かを判断する（ステップ808）。

【0084】すなわち、アルファベットの小文字{g、

$y, j$  のいずれかの文字の文字コードであるか否かを判断する。

【0085】ここで、いずれかの文字コードである場合は、その文字のバウンディングボックス  $BB_{ox}$  の底辺がライン  $LN_8$  に接するように、Y方向への移動量を求める（ステップ809）。

【0086】ステップ809を終了した場合、ステップ808において該当する文字の文字コードでない場合は、処理すべき文字は存在するか否かを判断し（ステップ810）、存在する場合には上記ステップ801に戻りこのステップ以降を実行し、一方、存在しない場合は処理を終了する。

【0087】こうして求められた移動量に基づいて文字を最適な位置に再配置することができる。

【0088】次に、再配置部70による文字の再配置処理について、図9に示すフローチャートを参照して説明する。

【0089】再配置部70は、キャラクタのX方向及びY方向の移動量として座標値を求める。ここで、X方向の移動量として検出部40によって求められた移動量とし、これを  $Xm_1$  と定義し、またY方向の移動量として配置演算部60によって求められた移動量とし、これを  $Ym_2$  と定義する（ステップ901）。

【0090】移動量（ $Xm_1, Ym_2$ ）を求めたら、バウンディングボックス  $BB_{ox}$  に関する各値を変更する。

【0091】すなわち、アセンダー及びディセンダーについてはそれぞれY方向に  $Ym_2$  だけ移動し、レフトベアリング及びライトベアリングについてはそれぞれX方向に  $Xm_1$  だけ移動する（ステップ902）。

【0092】次に、再配置部70は、対象の文字パターンがビットマップで表現されているか否かを判断する（ステップ903）。

【0093】ビットマップであれば、ドットの存在する座標点（ $x_i, y_i$ ）を、 $Xm_1, Ym_2$  だけ移動する。つまり、 $n$  個のドットの座標点を（ $x_i, y_i$ ） $(i = 1, 2, \dots, n \text{ (整数)})$  とすれば、 $x_i = x_i + Xm_1, y_i = y_i + Ym_2$  の各式を演算して得られる座標にドットを移動させる（ステップ904）。

【0094】ステップ903においてビットマップでない場合（この場合はアウトラインということになる）は、 $n$  個の制御点の座標点を（ $x_i, y_i$ ） $(i = 1, 2, \dots, n \text{ (整数)})$  とすれば、 $x_i = x_i + Xm_1, y_i = y_i + Ym_2$  の各式を演算して得られる座標にドットを移動させる（ステップ905）。

【0095】ステップ904を終了した場合、ステップ905を終了した場合は、処理すべき文字が存在するか否かを判断する（ステップ906）。

【0096】ここで、存在する場合には上記ステップ901に戻りこのステップ以降を実行し、一方、存在しな

い場合は処理を終了する。

【0097】以上の処理により、英数文字パターンの位置が最適な位置に自動修正されたこととなる。

【0098】次に本発明の第2の実施例を、図10乃至図17を参照して説明する。

【0099】図10は本発明に係る文字パターン自動修正装置の第2の実施例を機能ブロック図で示したものである。

【0100】同図において、文字パターン自動修正装置は、文字フォントに対応する文字コードを入力する入力装置1010と、作成された文字パターンを格納しているフォントメモリ1020と、指定された文字コードに対応する文字パターンをフォントメモリ1020から読み込む読込部1030と、予め設定された標準書体の文字パターンを格納している標準書体フォントメモリ1040と、読込部1030によって読み込まれた文字パターンの位置を検出すると共に、標準書体フォントメモリ1040からその文字パターンの文字コードと対応する標準書体の文字パターンを読み込んでこの文字パターンの位置を検出する検出部1050と、作成された文字パターンの位置と標準書体の文字パターンの位置に基づいて作成された文字の移動すべき位置を演算する配置演算部1060と、配置演算部1060により求められた位置に文字を再配置する再配置部1070とを有して構成されている。

【0101】フォントメモリ20には、JISの文字コードと、その文字コードに対応する文字パターン（アウトライン又はビットマップ）及びメトリクス情報からなるパターン情報とが格納されている。ここには、例えば図11(a)、(b)に示すような作成された文字パターンが格納される。ここでは、前提条件として文字のデザインは正しいものとする。

【0102】標準書体フォントメモリ1040には、あらゆる書体の模範となる標準書体の文字パターンが格納されており、具体的には、上記同様に、JISの文字コードと、その文字コードに対応する文字パターン（アウトライン又はビットマップ）及びメトリクス情報からなるパターン情報とが格納されている。ここには、例えば図12に示すような標準書体の文字パターンが格納されている。ここでは、例えばJIS X0208の6877文字が格納されているものとし、また前提条件として標準書体のデザイン、文字の並びとともに正しいデータであるものとする。

【0103】ここで、上記パターン情報について、図13に示す“A”を例にとって説明する。

【0104】メトリクス情報は、“A”という文字マスクと、図13中点線で囲まれたバウンディングボックス  $BB_{ox}$ （文字パターンに接する最大矩形枠）の情報と、文字の送り幅であるエスキープメント  $W$  の情報とを有して構成されている。バウンディングボックス  $BB_{ox}$

13

x情報は、原点(0, 0)からのオフセット値であるアセンダーA(正の値)、ディセンダーD(負の値)、ライトペアリングR(正の値)及びレフトペアリングL(負の値)の4つの情報から構成されている。このようなメトリクス情報は、作成された文字パターン及び標準書体の文字パターンそれぞれについて定義される。

【0105】なお、この実施例では、上述した記憶手段は標準書体フォントメモリ1040を有して構成されており、上述した検出手段は検出部1050を有して構成されており、上述した演算手段は配置演算部1060を有して構成されており、再配置手段は再配置部1070を有して構成されている。

【0106】係る構成において、文字パターン自動修正装置の文字パターンの位置修正処理について説明する。

【0107】入力装置1010から、文字“A”、“B”、“あ”、“い”的各文字にそれぞれ対応する文字コードが順次入力されると、読出部1030は、フォントメモリ1020からフォントファイルを読み込み、そのファイル内から、それらの文字コードに対応する文字パターンを読み込む。この結果、図11(a)に示すような文字パターンの並びが得られることとなる。しかし、これらの文字パターンは、図12に示すように最適な並びでないことが分かる。

【0108】そこで、それらの文字パターンの位置を検査し、最適な位置に修正する必要があるので、次に、本発明に係る文字パターンの位置の検査及び修正について説明する。

【0109】ここでは、最初の文字“A”を例にとって説明することにし、その文字の位置を検査し修正する処理過程を図14乃至図16を参照して説明する。

【0110】読込部1030でフォントファイルを読み込み、文字コードによって文字を判別する。この場合、JISコードになっているため大文字の“A”は0x2341(JIS3区33点)である。すなわち、読込部1030は、入力装置1010からの文字コード0x2341を受け取ると、該文字コードに基づいて大文字の“A”であると判定し、更に、その文字の文字パターン及びメトリクス情報をフォントファイルから読み取ると共に、読み取ったこれらの情報及び文字コードを検出部1050に渡す。

【0111】検出部1050では、文字“A”的バウンディングボックス情報(アセンダー、ディセンダー、レフトペアリング、ライトペアリング)から、図14に示されるy0、y1、x0、x1を検出する。図14において、MXは文字が登録されるため設定された領域(ボディーサイズ)の縦の長さ、MYは前記領域(ボディーサイズ)の横の長さを示している。

【0112】すなわち、

y0はディセンダーの値の絶対値と同一である。

(y0=- (ディセンダーの値))

14

y1はMYの値からアセンダーの値を減算して得られる値である。

(y1=MYの値-アセンダーの値)

x0はレフトペアリングの値の絶対値と同一である。

(x0=- (レフトペアリングの値))

x1はMXの値からライトペアリングの値を減算して得られる値である。

(x1=MXの値-ライトペアリングの値)

このようにして求められたy0、y1、x0、x1の値は作成された文字パターンに関する位置情報として定義される。

【0113】次に検出部1050は、標準書体の文字“A”、すなわち文字コード0x2341の文字を標準書体フォントメモリ1040から読み出す。ここで、標準書体の文字“A”は図15に示すような内容になっているものとする。図15において、MX0はボディーサイズの縦の長さ、MY0はボディーサイズの横の長さを示している。この標準文字パターンについても、標準書体の文字“A”的バウンディングボックス情報から、図15に示されるya、yb、xa、xbを検出する。これらの値は上記同様にして求めることができる。

【0114】すなわち、

ya=MY0の値-アセンダーの値

yb=- (ディセンダーの値)

xa=- (レフトペアリングの値)

xb=MX0の値-ライトペアリングの値

このようにして求められたya、yb、xa、xbの値は標準書体の文字パターンに関する位置情報として定義される。

【0115】作成された文字パターン及び標準書体の文字パターンに関する位置情報は、検出部1050から配置演算部1060に出力される。

【0116】配置演算部1060によって、作成された文字パターンに関する位置情報と標準書体の文字パターンに関する位置情報とを比較し、この比較結果に基づいて最適な位置への移動量を演算し、更に再配置部1070によって、その移動量に基づいて、作成された文字パターンを移動させることにより、結果として図16に示すような文字パターンが得られることとなる。

【0117】次に、配置演算部1060による移動量の演算処理について、図17に示すフローチャートを参照して説明する。

【0118】配置演算部1060は、検出部1050からの作成された文字パターンに関する位置情報(y0、y1、x0、x1)を読み込むと共に、A=(x0+x1)と定義し、またB=(y0、y1)と定義する(ステップ1701)。また検出部1050からの標準書体の文字パターンに関する位置情報(ya、yb、xa、xb)を読み込む(ステップ1702)。

50 【0119】次に、配置演算部1060は、x0:x1

$=x a : x b$  の式が成立するか否かを調べ（ステップ1703）、この式が成立する場合はX方向の位置は正しいことになるので、次にY方向についての移動量を求めることになる。

【0120】一方ステップ1703において上記式が不\*

$$x 0 \text{ new} : x 1 \text{ new} = x a : x b \quad \dots (6)$$

の式が成立するように、 $x 0 \text{ new}$ 及び $x 1 \text{ new}$ の各値を算出する。

【0122】この場合、 $x 0 + x 1 = x 0 \text{ new} + x 1$ ※

$$x 1 \text{ new} = A - x 0 \text{ new} \quad \dots (7)$$

の式が得られる。

【0123】また、この式（7）の関係から、上記式★

$$x 0 \text{ new} : (A - x 0 \text{ new}) = x a : x b \quad \dots (8)$$

の式に変形することができる。

【0124】更に式（8）においてA、 $x a$ 、 $x b$ は定☆

$$x 0 \text{ new} = (A \times x a) / (x a + x b) \quad \dots (9)$$

の式が得られる。

【0125】この式（9）を演算することにより $x 0 \text{ new}$ の値すなわちX方向の最適な位置が得られる。

【0126】そして配置演算部1060は、上記式（9）を演算して得られた $x 0 \text{ new}$ の値から $x 0$ を減算した値をX方向の移動量とする（X方向の移動量= $x 0 \text{ new} - x 0$ ）（ステップ1705）。

【0127】またステップ1705を終了した場合、上記ステップ1703において式が成立する場合は、配置◆

$$y 0 \text{ new} : y 1 \text{ new} = y b : y a \quad \dots (10)$$

の式が成立するように、 $y 0 \text{ new}$ 及び $y 1 \text{ new}$ の各値を算出する。

【0130】この場合、 $y 0 + y 1 = y 0 \text{ new} + y 1$ ※

$$y 1 \text{ new} = B - y 0 \text{ new} \quad \dots (11)$$

の式が得られる。

【0131】また、この式（11）の関係から、上記式★

$$y 0 \text{ new} : (B - y 0 \text{ new}) = y b : y a \quad \dots (12)$$

の式に変形することができる。

【0132】さらに式（12）においてB、 $y a$ 、 $y b$ ★

$$y 0 \text{ new} = (B \times y b) / (y a + y b) \quad \dots (12)$$

の式が得られる。

【0133】この式（12）を演算することにより $y 0 \text{ new}$ の値すなわちY方向の最適な位置が得られる。

【0134】そして配置演算部1060は、上記式（12）を演算して得られた $y 0 \text{ new}$ の値から $y 0$ を減算した値をY方向の移動量とする（Y方向の移動量= $y 0 \text{ new} - y 0$ ）（ステップ1708）。

【0135】ステップ1708を終了した場合、上記ステップ1703において式が成立した場合は処理を終了する。

【0136】以上の処理で、X方向及びY方向の移動量が算出されることになる。なお、ステップ1703でYES、更にステップ1706でYESの場合は、作成された文字パターンと標準書体の文字パターンのX方向の

\*成立の場合は、X方向の文字の位置は最適ではないので、その文字を最適な位置に移動すべく移動量を算出する（ステップ1704）。

【0121】すなわち、

$$\dots (6)$$

※ $\text{new} = M X$ の値 - (ライトペアリングの値 + レフトペアリングの値) = A (定数) の関係から、

$$\dots (7)$$

★ (6) は、

$$\dots (8)$$

☆数であるので、式（8）からは、

◆演算部1060は、 $y 0 : y 1 = y b : y a$ の式が成立するか否かを調べ（ステップ1706）、この式が成立する場合はY方向の位置は正しいことになる。

【0128】一方ステップ1706において上記式が不成立の場合は、Y方向の文字の位置は最適ではないので、その文字を最適な位置に移動すべく移動量を算出する（ステップ1707）。

【0129】すなわち、

$$\dots (10)$$

\* $\text{new} = M Y$ の値 - (アセンダーの値 + ディセンダーの値) = B (定数) の関係から、

$$\dots (11)$$

※ (10) は、

$$\dots (12)$$

★は定数であるので、式（12）からは、

位置及びY方向の位置が一致していることとなり、作成された文字パターンは最適な位置に存在していることになる。

【0137】上述したように配置演算部1060によって求められたX方向及びY方向の移動量に基づいて、作成された文字パターンを最適な位置に移動（再配置）させることができる。

【0138】この移動処理（再配置）は再配置部1070によって行われることになるが、再配置部1070による再配置処理は、第1の実施例の図9に示される処理手順と同様なので、ここではその説明を省略する。

【0139】

【発明の効果】以上説明したように第1の発明によれば、英数文字パターンの原点に対する当該英数文字の外

17

接四角形の位置関係を示すメトリクス情報に基づいて、その英数文字の位置が最適か否かを判定し、この判定の結果、最適な位置に存在していない英数文字については、上記メトリクス情報に基づいて演算した移動量に従って再配置されるので、英数文字（例えばアスキーコード）に対応する英数文字パターンの位置を最適な位置に修正することのできる文字パターン自動修正装置を提供することができる。従って、正確な文字並びの文書を作成することができる。

【0140】また第2の発明によれば、英数文字パターンを含む文字パターンの原点に対する当該文字の外接四角形の位置関係を示すメトリクス情報に基づいて、その文字の位置が最適か否かを判定し、この判定の結果、最適な位置に存在していない文字については、上記メトリクス情報と標準書体の文字パターンに関するメトリクス情報とに基づいて演算した移動量に従って再配置されるので、英数文字、ひら仮名などの文字（例えばJISの文字コード）に対応する文字パターンの位置を最適な位置に修正することのできる文字パターン自動修正装置を提供することができる。従って、正確な文字並びの文書を作成することができる。

【図面の簡単な説明】

【図1】本発明に係る文字パターン自動修正装置の第1の実施例を示す機能ブロック図。

【図2】英数文字パターンのメトリクス情報を説明するための図。

【図3】標準的な英数文字パターンの並びを説明するための図。

【図4】修正前の英数文字パターンの並びを説明するための図。

【図5】修正前の英数文字パターンの位置を最適な位置

10

20

30

18

に修正するための処理過程を説明するための図。

【図6】ライン演算部によるオーバハングの算出処理の処理手順を示すフローチャート。

【図7】ライン演算部による英数文字の並び線（ライン）の決定処理の処理手順を示すフローチャート。

【図8】配置演算部による文字パターンの移動量算出処理の処理手順を示すフローチャート。

【図9】再配置部による文字パターンの再配置処理の処理手順を示すフローチャート。

【図10】本発明に係る文字パターン自動修正装置の第2の実施例を示す機能ブロック図。

【図11】フォントメモリに格納される作成された文字パターンの一例を示す図。

【図12】標準書体フォントメモリに格納される標準書体の文字パターンの一例を示す図。

【図13】文字パターンのメトリクス情報を説明するための図。

【図14】修正前の作成された文字パターンに関する位置情報の算出を説明するための図。

【図15】標準書体の文字パターンに関する位置情報の算出を説明するための図。

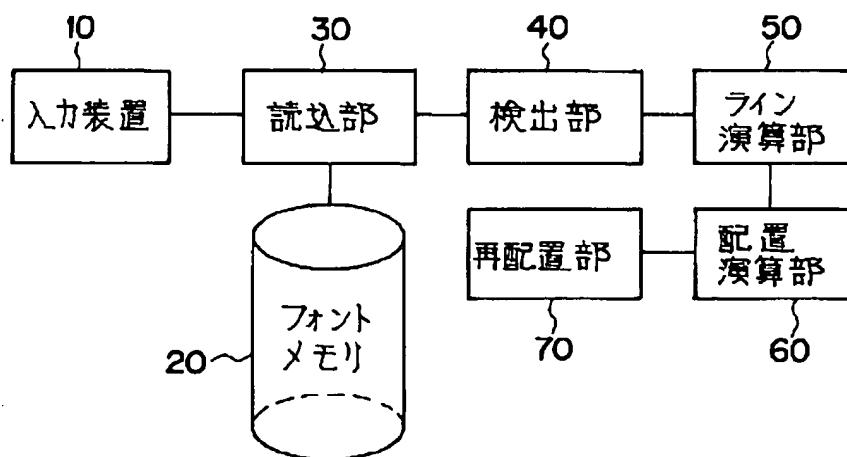
【図16】修正後の作成された文字パターンに関する位置情報の算出を説明するための図。

【図17】配置演算部による文字パターンの移動量算出処理の処理手順を示すフローチャート。

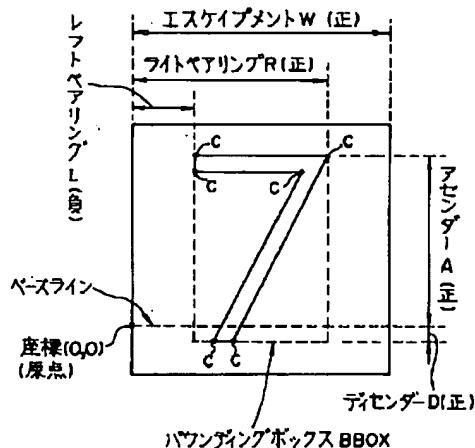
【符号の説明】

10、1010…入力装置、20、1020…フォントメモリ、30、1030…読み部、40、1050…検出部、50…ライン演算部、60、1060…配置演算部、70、1070…再配置部、1040…標準書体フォントメモリ。

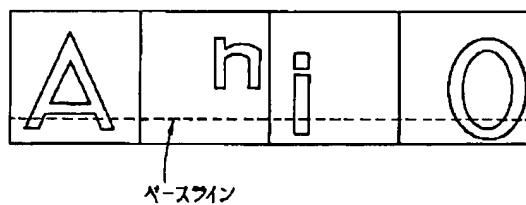
【図1】



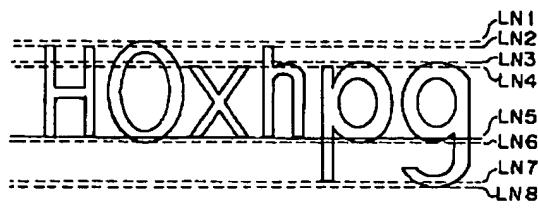
【図2】



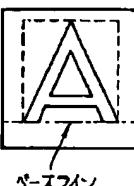
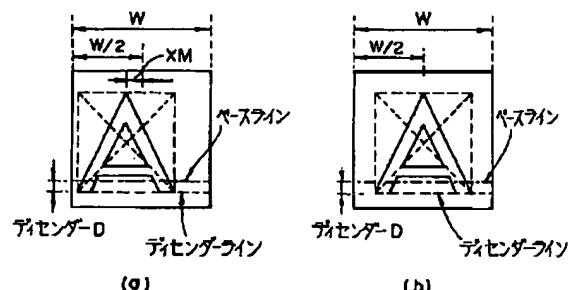
【図4】



【図3】

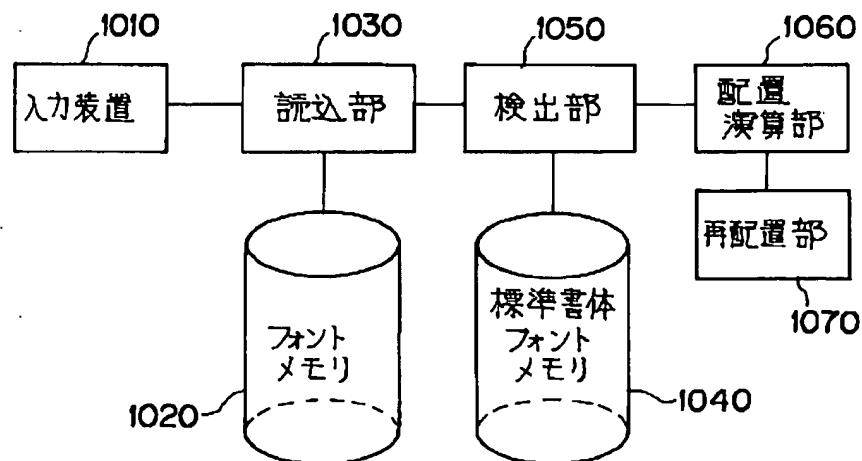


【図5】

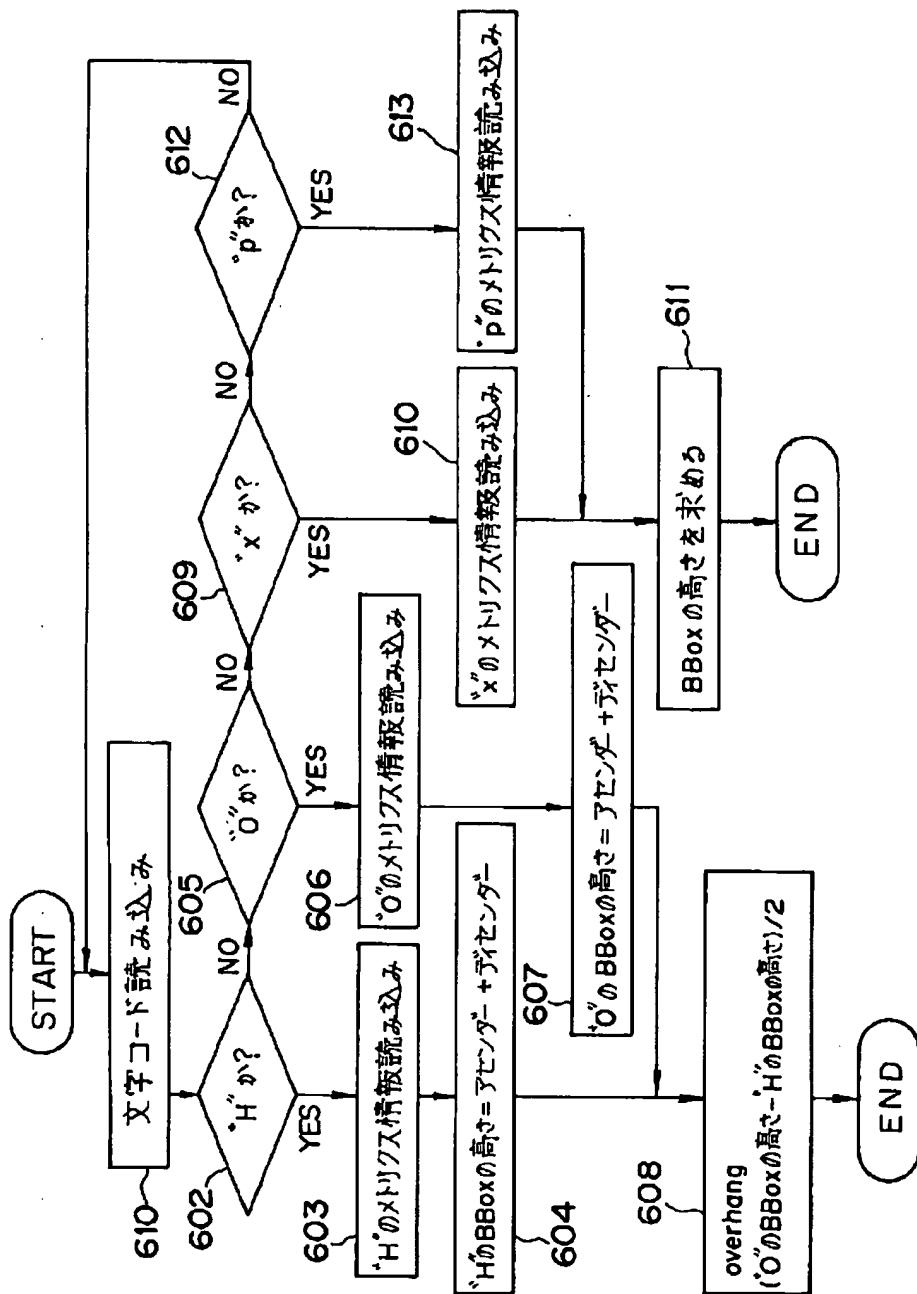


【c】

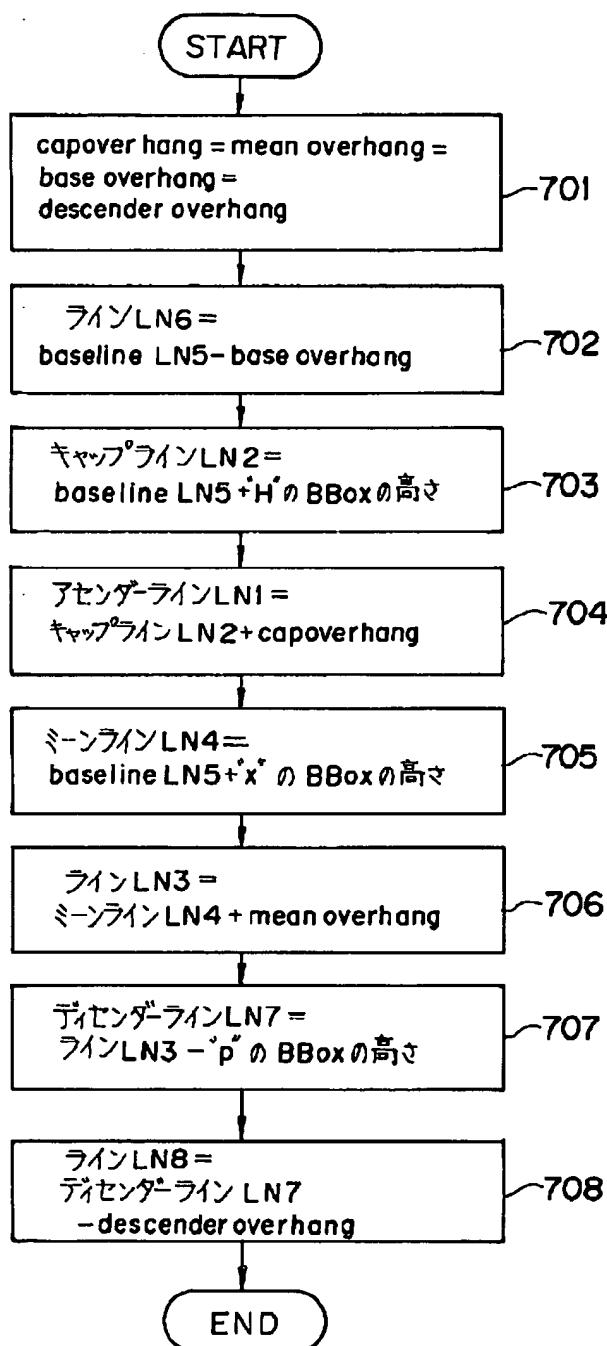
【図10】



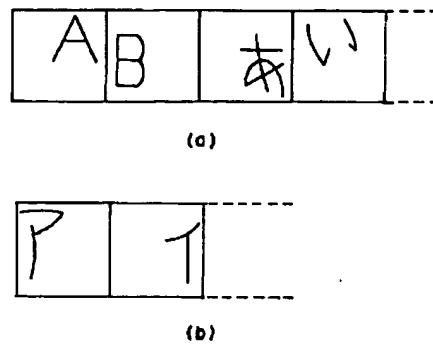
【図6】



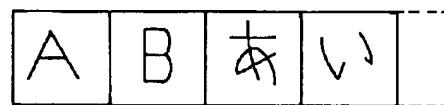
【図7】



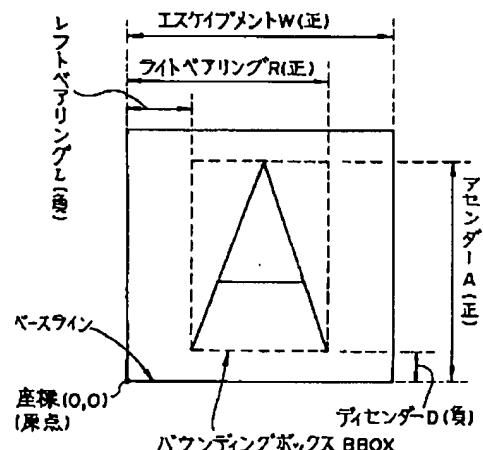
【図11】



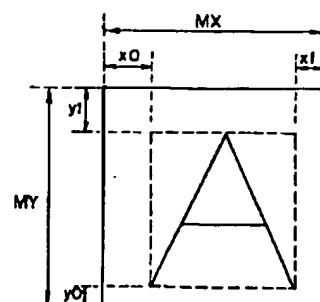
【図12】



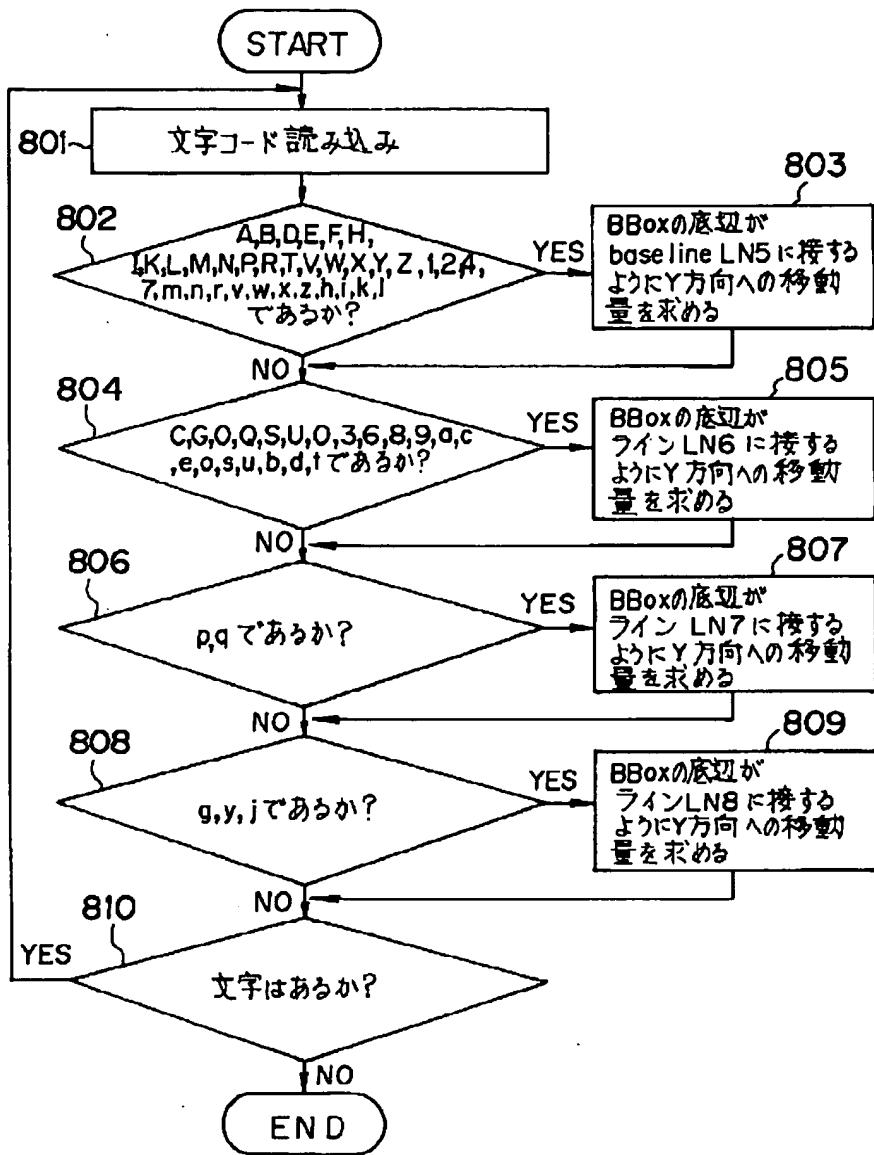
【図13】



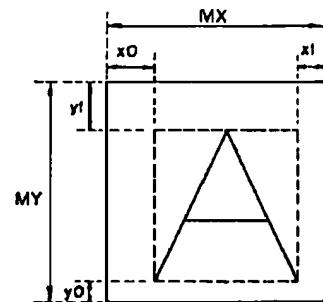
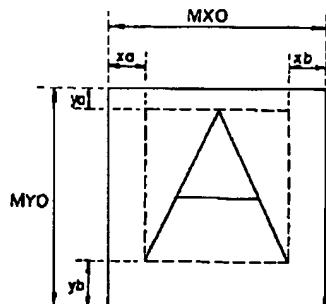
【図14】



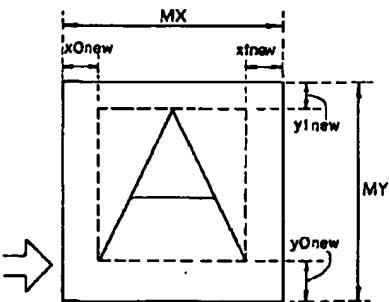
【図8】



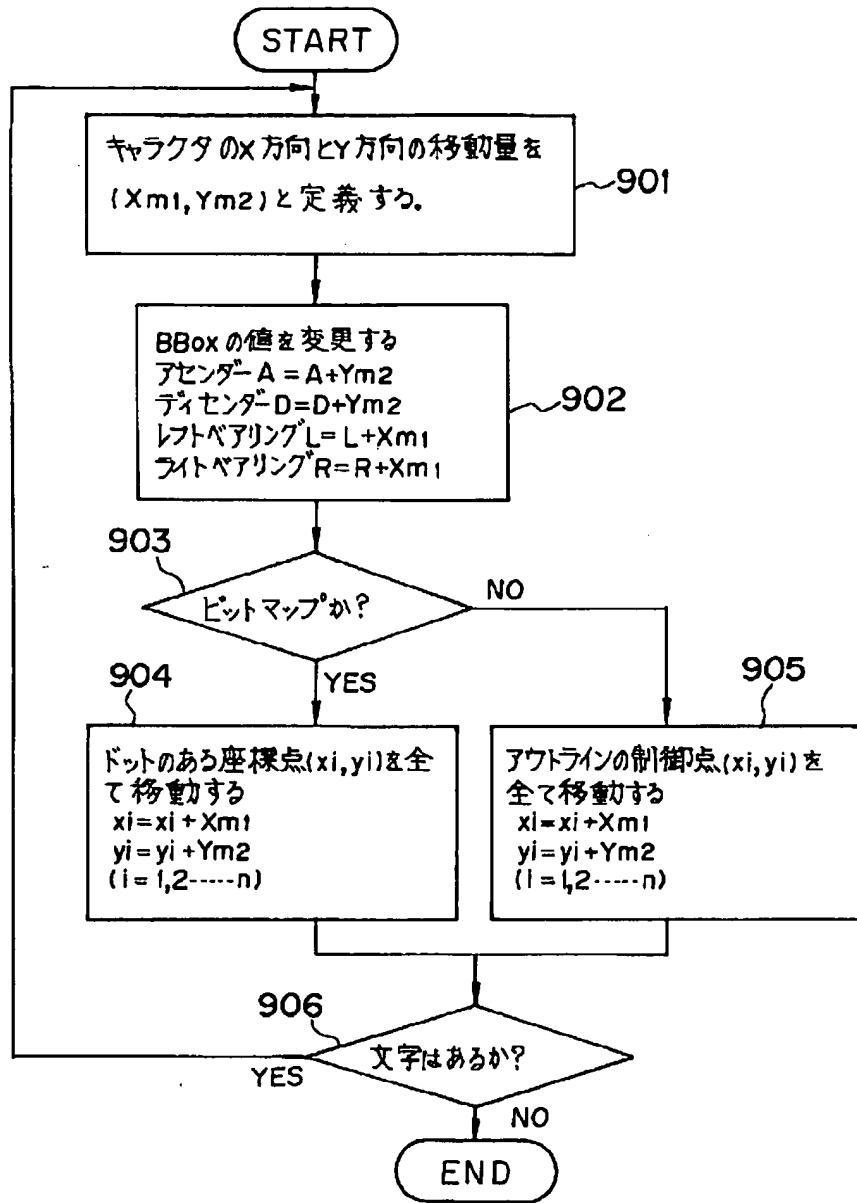
【図15】



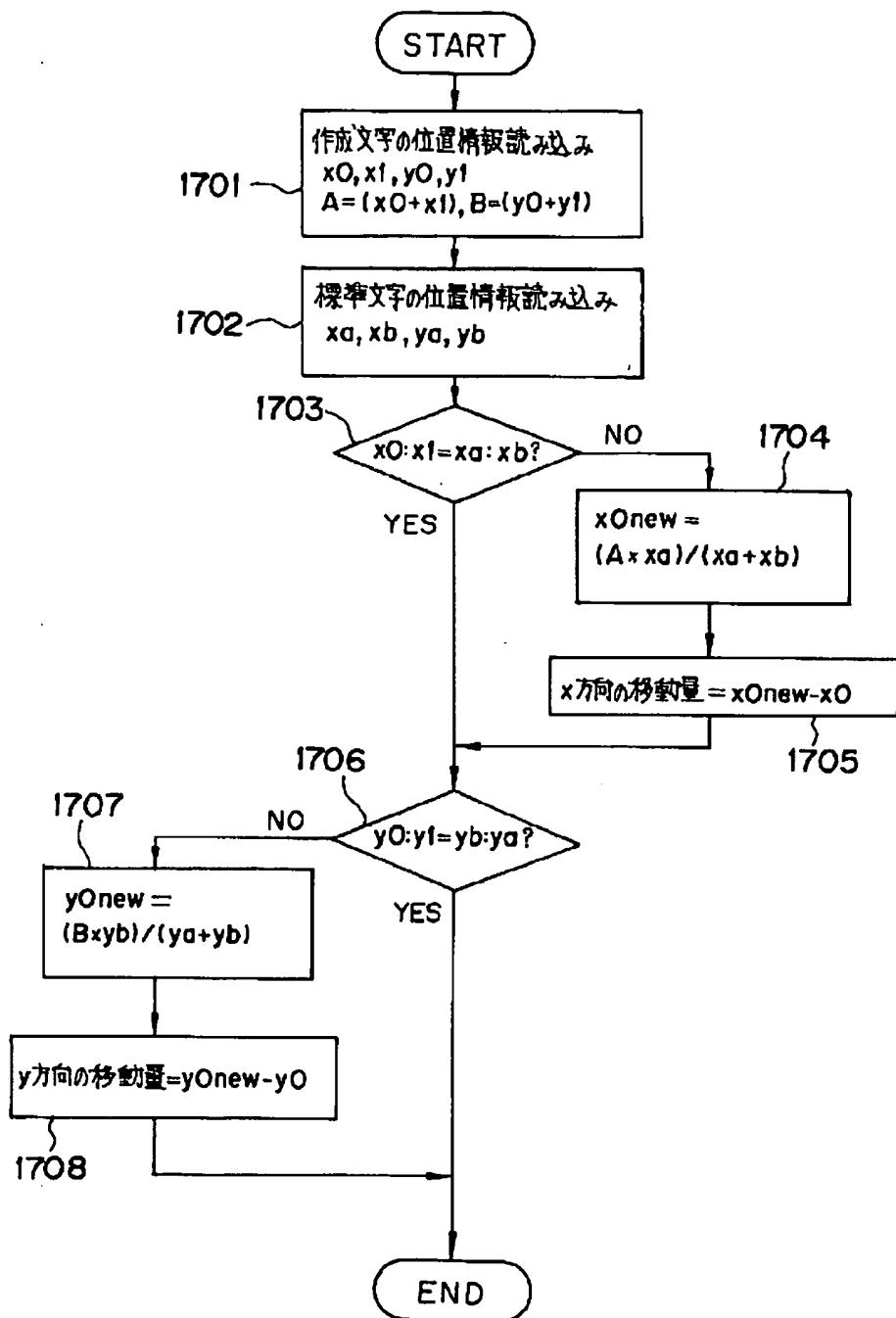
【図16】



【図9】



【図17】



フロントページの続き

(51) Int.Cl. 6

B 41 J 2/485

G 06 F 15/20

識別記号 庁内整理番号

F 1

技術表示箇所

5 3 4 P 7343-5L